SYNOPSIS

ON

“Desktop Voice Assistant”

Submitted in

Partial Fulfillment of requirements for the Award of Degree

*of*

Bachelor of Technology

*In*

Computer Science and Engineering

By

**( Project Id: 2C16 )**

# Divyanshi Gupta (1901640100118)

# Shatakchhi Vishnoi (1901640100255)

Under the supervision of

**Garima Verma**

**(Assistant Professor)**



**Pranveer Singh Institute of Technology**.

 Kanpur - Agra - Delhi National Highway - 2

Bhauti -Kanpur - 209305.

(Dr. A.P.J. Abdul Kalam Technical University)

**Synopsis**

1. Introduction

What is voice Assistant?

A voice assistant is a digital assistant that uses *voice* recognition, speech synthesis, and natural language processing (NLP) to provide a service through a particular application.

Examples of voice Assistant

Amazon’s Alexa, Google Home, Apple’s Siri, and Microsoft’s Cortana are popular examples of voice assistance that use artificial intelligence (AI).

**2. Project Objective**

What can this A.I. assistant do for you?

* It can send emails on your behalf.
* It can play music for you.
* It can do Wikipedia searches for you.
* It is capable of opening websites like Google, Youtube, etc., in a web browser.
* It is capable of opening your code editor or IDE with a single voice command.

3. Feasibility Study:

Feasibility study can help you determine whether or not you should proceed with your project. It is essential to evaluate cost and benefit. It is essential to evaluate cost and benefit of the proposed system. Five types of feasibility study are taken into consideration.

1.Technical feasibility:

It includes finding out technologies for the project, both hardware and software. For voice assistant, user must have microphone to convey their message and a speaker to listen when system speaks. These are very cheap now adays and everyone generally possess them. Besides, system needs internet connection.While using them, make sure you have a steady internet connection. It is also not anissue in this era where almost every home or office has Wi-Fi.

2.Operational feasibility:

It is the ease and simplicity of operation of proposed system.System does not require any special skill set for users to operate it. In fact, it is designed to be used by almost everyone. Kids who still don’t know to write can readout problems for system and get answers.

3.Economical feasibility:

 Here, we find the total cost and benefit of the proposed system over current system. For this project, the main cost is documentation cost. User also would have to pay for microphone and speakers. Again, they are cheap and available. As far as maintenance is concerned, voice assistants won’t cost too much.

4.Organizational feasibility:

This shows the management and organizational structure of the project. This project is not built by a team. The management tasks are all to

 becarried out by a single person. That won’t create any management issues and will increase the feasibility of the project.

5.Cultural feasibility:

It deals with compatibility of the project with cultural environment. Virtual assistant is built in accordance with the general culture. This project is technically feasible with no external hardware requirements. Also it is simple in operation and does not cost training or repairs. Overall feasibility study of  the project reveals that the goals of the proposed system are achievable. Decision is  taken to proceed with the project.

4. Methodology/ Planning of work

#### Using VS Code

For making this voice assistant we have used VS Code IDE. Start a new project and make a file called jarvis.py.

#### Defining Speak Function

The first and foremost thing for an A.I. assistant is that it should be able to speak. To make our J.A.R.V.I.S. talk, we will make a function called **speak().**This function will take audio as an argument, and then it will pronounce it.

def speak(audio):

pass #For now, we will write the conditions later.

Now, the next thing we need is audio. We must supply audio so that we can pronounce it using the speak() function we made. We are going to install a module called **pyttsx3.**

##### What is pyttsx3?

* A python library that will help us to convert text to speech. In short, it is a text-to-speech library.
* It works offline, and it is compatible with Python 2 as well as Python 3.

**Installation:**

pip install pyttsx3

After successfully installing pyttsx3, import this module into your program.

**Usage:**

import pyttsx3

engine = pyttsx3.init('sapi5')

voices= engine.getProperty('voices') #getting details of current voice

engine.setProperty('voice', voice[0].id)

##### What is sapi5?

* Microsoft developed speech API.
* Helps in synthesis and recognition of voice.

##### What Is VoiceId?

* Voice id helps us to select different voices.
* voice[0].id = Male voice
* voice[1].id = Female voice

#### Writing Our speak() Function :

We made a function called speak() at the starting of this tutorial. Now, we will write our speak() function to convert our text to speech.

def speak(audio):

engine.say(audio)

engine.runAndWait() #Without this command, speech will not be audible to us.

#### Creating Our main() function:

We will create a main() function, and inside this main() Function, we will call our speak function.

**Code:**

if \_\_name\_\_=="\_\_main\_\_" :

speak("Coding is life")

Whatever you will write inside this speak() function will be converted into speech. Congratulations! With this, our J.A.R.V.I.S. has its own voice, and it is ready to speak.

#### Defining Wish me Function :

Now, we will make a **wishme()**function, that will make our J.A.R.V.I.S. wish or greet the user according to the time of computer or pc. To provide current or live time to A.I., we need to import a module called datetime. Import this module to your program, by:

import datetime

Now, let's start defining the **wishme()** function:

def wishme():

hour = int(datetime.datetime.now().hour)

Here, we have stored the current hour or time integer value into a variable named hour. Now, we will use this hour value inside an if-else loop.

#### Defining Take command Function :

The next most important thing for our A.I. assistant is that it should take command with the help of the microphone of the user's system. So, now we will make a **takeCommand()**function.  With the help of the takeCommand() function, our A.I. assistant will return a string output by taking microphone input from the user.

 Before defining the takeCommand() function, we need to install a module called **speechRecognition.**Install this module by:

pip install speechRecognition

After successfully installing this module, import this module into the program by writing an import statement.

import speechRecognition as sr

Let's start coding the takeCommand() function :

def takeCommand():

#It takes microphone input from the user and returns string output

r = sr.Recognizer()

with sr.Microphone() as source:

print("Listening...")

r.pause\_threshold = 1

audio = r.listen(source)

We have successfully created our takeCommand() function. Now we are going to add a try and except block to our program to handle errors effectively.

try:

print("Recognizing...")

query = r.recognize\_google(audio, language='en-in') #Using google for voice recognition.

print(f"User said: {query}\n") #User query will be printed.

except Exception as e:

# print(e)

print("Say that again please...") #Say that again will be printed in case of improper voice

return "None" #None string will be returned

return query

#### Coding logic of Jarvis

 Now, we will develop logic for different commands such as Wikipedia searches, playing music, etc.

##### Defining Task 1: To search something on Wikipedia

 To do Wikipedia searches, we need to install and import the Wikipedia module into our program. Type the below command to install the Wikipedia module :

pip install wikipedia

 After successfully installing the Wikipedia module, import it into the program by writing an import statement.

if \_\_name\_\_ == "\_\_main\_\_":

wishMe()

while True:

# if 1:

query = takeCommand().lower() #Converting user query into lower case

# Logic for executing tasks based on query

if 'wikipedia' in query: #if wikipedia found in the query then this block will be executed

speak('Searching Wikipedia...')

query = query.replace("wikipedia", "")

results = wikipedia.summary(query, sentences=2)

speak("According to Wikipedia")

print(results)

speak(results)

In the above code, we have used an if statement to check whether Wikipedia is in the search query of the user or not. If Wikipedia is found in the user's search query, then two sentences from the summary of the Wikipedia page will be converted to speech with the speak function's help.

##### Defining Task 2: To open YouTube site in a web-browser

 To open any website, we need to import a module called webbrowser. It is an in-built module, and we do not need to install it with a pip statement; we can directly import it into our program by writing an import statement.

Code:

elif 'open youtube' in query:

webbrowser.open("youtube.com")

Here, we are using the elif loop to check whether Youtube is in the user's query. Let' suppose the user gives a command as "J.A.R.V.I.S., open youtube." So, open youtube will be in the user's query, and the elif condition will be true.

##### Defining Task 3: To open Google site in a web-browser

elif 'open google' in query:

webbrowser.open("google.com")

We are opening Google in a web-browser by applying the same logic that we used to open youtube.

##### Defining Task 4: To play music

To play music, we need to import a module called os. Import this module directly with an import statement.

elif 'play music' in query:

music\_dir = 'D:\\Non Critical\\songs\\Favorite Songs2'

songs = os.listdir(music\_dir)

print(songs)

os.startfile(os.path.join(music\_dir, songs[0]))

In the above code, we first opened our music directory and then listed all the songs present in the directory with the os module's help. With the help of os.startfile, you can play any song of your choice. I am playing the first song in the directory. However, you can also play a random song with the help of a random module. Every time you command to play music, J.A.R.V.I.S. will play any random song from the song directory.

##### Defining Task 5: To know the current time

elif 'the time' in query:

strTime = datetime.datetime.now().strftime("%H:%M:%S")

speak(f"Sir, the time is {strTime}")

In the above, code with using datetime() function and storing the current or live system into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument in speak function. Now, the time string will be converted into speech.

##### Defining Task 6: To open the VS Code Program

elif 'open code' in query:

codePath = "C:\\Users\\CodingTut\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"

os.startfile(codePath)

To open the VS Code or any other application, we need the code path of the application.

**Steps to get the code path of the application:**

**Step 1:**Open the file location.

**Step 2:**Right-click on the application and click on properties.

**Step 3:**Copy the target from the target section.

After copying the target of the application, save the target into a variable. Here, we are saving the target into a variable called codePath, and then we are using the os module to open the application.

##### Defining Task 7: To send Email

To send an email, we need to import a module called smtplib.

**What is smtplib?**

* Simple Mail Transfer Protocol (SMTP) is a protocol that allows us to send emails and to route emails between mail servers. An instance method called **sendmail**is present in the SMTP module. This instance method allows us to send an email.  It takes 3 parameters:
* **The sender:** Email address of the sender.
* **The receiver:**T Email of the receiver.
* **The *message:*** A string message which needs to be sent to one or more than one recipient.

##### Defining Send email function :

We will create a **sendEmail()**function, which will help us send emails to one or more than one recipient.

def sendEmail(to, content):

server = smtplib.SMTP('smtp.gmail.com', 587)

server.ehlo()

server.starttls()

server.login('youremail@gmail.com', 'your-password')

server.sendmail('youremail@gmail.com', to, content)

server.close()

In the above code, we are using the SMTP module, which we have already discussed above.

Also 'enable the less secure apps' feature in your Gmail account. Otherwise, the sendEmail function will not work properly.

#### Calling sendEmail() function inside the main() function:

elif 'email to harry' in query:

try:

speak("What should I say?")

content = takeCommand()

to = "universakkid@gmail.com"

sendEmail(to, content)

speak("Email has been sent!")

except Exception as e:

print(e)

speak("Sorry ma’am... I am not able to send this email")

We are using the try and except block to handle any possible error while sending emails.

#### Recapitulate

1. First of all, we have created a **wishme()**function that gives the greeting functionality according to our A.I system time.
2. After wishme() function, we have created a **takeCommand()**function, which helps our A.I to take command from the user. This function is also responsible for returning the user's query in a string format.
3. We developed the code logic for opening different websites like google, youtube, and stack overflow.
4. Developed code logic for opening VS Code or any other application.
5. At last, we added functionality to send emails.

#### With this, we have successfully made our very first virtual assistant.

5. Facilities required for proposed work

5.1 Hardware Requirements

* Computer with a sound card
* Microphone

5.2 Software Requirements

* Code Editor: Visual Studio Code
* Python 3.9

**6. Coding**

**import pyttsx3**

**import datetime**

**import wikipedia**

**import webbrowser**

**import random**

**import os**

**import smtplib**

**import speech\_recognition as sr**

**engine=pyttsx3.init('sapi5')**

**voices=engine.getProperty('voices')**

**engine.setProperty('voice',voices[0].id)**

**def speak(audio):**

**engine.say(audio)**

**engine.runAndWait()**

**def wishme():**

**hour=int (datetime.datetime.now().hour)**

**if hour>=0 and hour<12:**

**speak("Good Morning!")**

**elif hour>=12 and hour<=17:**

**speak("Good Afternoon!")**

**else:**

**speak("Good Evening!")**

**speak("I am V A . please tell me how may i help you?")**

**def takeCommand():**

**#It takes microphone input from the user and returns string output**

**r = sr.Recognizer()**

**with sr.Microphone() as source:**

**print("Listening...")**

**r.pause\_threshold = 1**

**r.energy\_threshold = 400**

**audio = r.listen(source)**

**try:**

**print("Recognizing...")**

**query = r.recognize\_google(audio, language='en-in')**

**print(f"User said: {query}\n")**

**except Exception as e:**

**# print(e)**

**print("Say that again please...")**

**return "None"**

**return query**

**def sendEmail(to,content):**

**server=smtplib.SMTP('smtp.gmail.com',587)**

**server.ehlo()**

**server.starttls()**

**server.login()**

**server.sendmail('vishnoishatakchhi@gmail.com',to,content)**

**server.close()**

**if \_name=="main\_":**

**wishme()**

**while True:**

**query=takeCommand().lower()**

**#logic for implementing task accordinb to query**

**if 'wikipedia' in query:**

**speak("searching wikipedia...")**

**query=query.replace("wikipedia","")**

**results=wikipedia.summary(query,sentences=2)**

**speak("According to wikipedia")**

**print(results)**

**speak(results)**

**elif 'open youtube' in query:**

**webbrowser.open('youtube.com')**

**elif 'open google' in query:**

**webbrowser.open('google.com')**

**elif 'open psit ' in query:**

**webbrowser.open('psit.ac.in')**

**elif 'open notepad' in query:**

**notepadpath='C:\\WINDOWS\\system32\\notepad.exe'**

**os.startfile(notepadpath)**

**elif 'open command prompt' in query:**

**os.system('start cmd')**

**elif 'open stack overflow' in query:**

**webbrowser.open('stackoverflow.com')**

**elif 'play music' in query:**

**music\_dir='E:\\favorite songs'**

**songs=os.listdir(music\_dir)**

**print(songs)**

**os.startfile(os.path.join(music\_dir,songs[random.randint(0,(len(songs)-1))])**

**elif 'the time' in query:**

**strTime=datetime.datetime.now().strftime("%H hours:%M minutes:%S seconds")**

**print(strTime)**

**speak(f"the time is{strTime}")**

**elif 'visual studio code' in query:**

**codePath="C:\\Users\\p\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"**

**os.startfile(codePath)**

**elif 'send email to khushi' in query:**

**try:**

**speak('what shoul i say?')**

**content=takeCommand()**

**to="khushigupta2000.kg@gmail.com"**

**sendEmail(to,content)**

**speak("Email has been sent!")**

**except Exception as e:**

**print(e)**

**speak('sorry my friend i am not able to send this email at the moment')**

**elif 'send email to Divyanshi' in query:**

**try:**

**speak('what shoul i say?')**

**content=takeCommand()**

**to="divyanshi12gupta@gmail.com"**

**sendEmail(to,content)**

**speak("Email has been sent!")**

**except Exception as e:**

**print(e)**

**speak('sorry my friend i am not able to send this email at the moment')**

**7. References**

1) www.youtube.com

2) github.com

3) www.wikipedia.org

4) www.geeksforgeeks.com

5) www.w3schools.com